Applications orientées données (NSY135)

4 – Modèle-Vue-Contrôleur (MVC)

Auteurs: Raphaël Fournier-S'niehotta et Philippe Rigaux
(philippe.rigaux@cnam.fr,fournier@cnam.fr)

Département d'informatique
Conservatoire National des Arts & Métiers, Paris, France

Plan du cours

S2 MVC : un exemple pratique

Présentation de l'exemple

- convertisseur de température, de l'unité Celsius vers l'unité Farenheit
- un seul contrôleur
- un seul objet métier
- deux vues
  - un formulaire permettant de saisir une valeur en Celsius
  - le résultat de la conversion de la valeur entrée en Fahrenheit

♣ Exemple volontairement simpliste, pour aller à l'essentiel

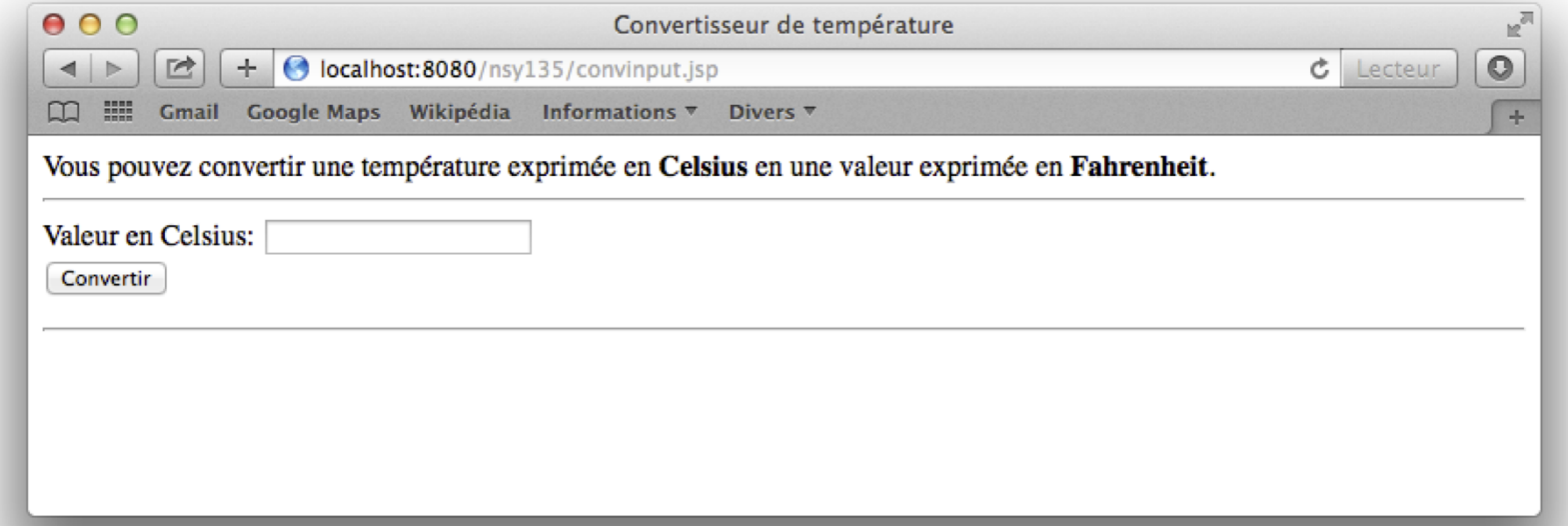
Java Server Pages

- technologie un peu ancienne, qui a beaucoup évolué
- idée : créer des pages HTML intégrant du code Java permettant d'avoir des parties dynamiques
- pas toujours élégant initialement, amélioré par :
  - intégration de balises spéciales dans le code
  - conventions de programmation et de nommage

Page d'accueil

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<html>

  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Convertisseur de température</title>
  </head>

  <body>
    Vous pouvez convertir une température exprimée en
    <b>Celsius</b> en une valeur exprimée en <b>Fahrenheit</b>.

    <hr />

    <form method="POST" action="${pageContext.request.contextPath}/convertisseur">

      Valeur en Celsius: <input type="text" size="20" name="celsius" /> <br />

      <input type="submit" value="Convertir" />
    </form>

    <hr />
  </body>

</html>
```

Formulaire du convertisseur



Convertisseur.java

- une servlet simple
- à associer à l'URL "/convertisseur"
- deux méthodes
  - doGet, affichage du formulaire
  - doPost, reçoit le paramètre du formulaire et conversion (action)

doGet

```
/**
 * Méthode Get: on affiche le formulaire
 */
 protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

    String maVue = "/convinput.jsp";

    RequestDispatcher dispatcher = getServletContext().getRequestDispatcher(maVue);

    dispatcher.forward(request,response);
}
```

- il y a délégation par le contrôleur de l'affichage à la vue (JSP)
- après sélection de la JSP à déclencher “convinput.jsp“ (dans “WebContent“)
- on lui transmet le flot d'exécution (forward)

Temperature

```java
package modeles;

/**
 * Une classe permettant d'obtenir une température en Celsius ou Farenheit.
 */
public class Temperature {

/**
 * La valeur, exprimée en degrés Celsius
 */
private double celsius;

/**
 * Le constructeur, prend des Celsius en paramètres
*/
public Temperature(double valeurCelsius)
{
  celsius = valeurCelsius;
}

/**
 * Pour obtenir la valeur en Celsius
 * @return Valeur de la température en Celsius
*/
public double getCelsius() {
  return celsius;
}

/**
 * Pour obtenir la valeur en Fahrenheit
 * @return Valeur de la température en Farenheit
*/
public double getFahrenheit() {
  return (celsius * 9/5) + 32;
}
}
```

Explications

- modèle très simple, représentation d'une température
- peut être fournie en Celsius ou Farenheit

Bonnes pratiques :

- aucune propriété n'est publique;
- des méthodes "set" et "get", au nommage normalisé, servent à lire/écrire les propriétés;
- les commentaires sont abondants et prêts à produire la javadoc

♣ cette classe est totalement indépendante du contexte Web et peut en fait être utilisée dans n'importe quelle application Java

Conversion

```
/**
* Méthode Post: on affiche la conversion
*/
protected void doPost(HttpServletRequest request, HttpServletResponse response)
  throws ServletException, IOException {
// On devrait récuperer la valeur saisie par l'utilisateur
String valCelsius = request.getParameter("celsius");

if (valCelsius.isEmpty()) {
  // Pas de valeur: il faudrait afficher un message, etc.
  valCelsius = "20";
}

// Action: appliquons le convertisseur. Espérons que valCelsius représente
// bien un nombre, sinon...
Temperature temp = new Temperature(Double.valueOf(valCelsius));
// Enregistrons l'objet dans la requête
request.setAttribute("temperature", temp);

// Transfert à la vue
String maVue = "/convoutput.jsp";
RequestDispatcher dispatcher =
    getServletContext().getRequestDispatcher(maVue);
dispatcher.forward(request,response);
}
```

Autre vue JSP

```
 <%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>

 <html>
  <head>
   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
   <title>Résultat de la conversion</title>
 </head>
 <body>

<p>Vous avez demandé la conversion en Fahrenheit de la valeur en
  Celsius ${requestScope.temperature.celsius}</p>
<p>
  <b>Et le résultat est: ${requestScope.temperature.fahrenheit}
    degrés Fahrenheit </b>!
</p>
 </body>
</html>
```

Affichage de la seconde JSP